



*Tecnologia em Análise e Desenvolvimento de Sistemas
4º Período*

Oficina de Integração
AN34E / N14

PROFESSOR

Diogo Cezar Teixeira Batista

DOCUMENTO DE PROJETO DE SOFTWARE

TECHFOOD: SISTEMA GERENCIADOR DE RESTAURANTE

Filipe Benevenuto Jandozo

João Victor Ferreira

Rafael Borduchi Amoroso

Cornélio Procópio
2017

SUMÁRIO

INTRODUÇÃO	5
CONTEXTO	5
JUSTIFICATIVA	6
PROPOSTA	6
ORGANIZAÇÃO DO DOCUMENTO	7
DESCRIÇÃO GERAL DO SISTEMA	8
OBJETIVOS	8
Objetivo Geral	8
OBJETIVOS ESPECÍFICOS	8
LIMITES E RESTRIÇÕES	8
DESCRIÇÃO DOS USUÁRIOS DO SISTEMA	9
DESENVOLVIMENTO DO PROJETO	10
TECNOLOGIAS E FERRAMENTAS	10
METODOLOGIA DE DESENVOLVIMENTO	10
CRONOGRAMA PREVISTO	11
REQUISITOS DO SISTEMA	12
REQUISITOS FUNCIONAIS	12
REQUISITOS NÃO-FUNCIONAIS	12
DIAGRAMAS DE CASOS DE USO	13
NÍVEL 0	13
GERENTE	14
FUNCIONÁRIO	15
ANÁLISE DO SISTEMA	16
MODELO DO BANCO DE DADOS	16
MODELO CONCEITUAL	16
MODELO LÓGICO	16
DICIONÁRIO DE DADOS	17

DIAGRAMA DE CLASSES	19
DIAGRAMA DE ATIVIDADES	20
IMPLEMENTAÇÃO	26
PROTÓTIPOS DE TELAS	26
TELA INICIAL	26
CARDÁPIO	27
LOGIN	27
USUÁRIO	28
GERENTE	28
FUNCIONÁRIO	28
7 CONSIDERAÇÕES FINAIS	31
8 REVISÃO BIBLIOGRÁFICA	32

1 INTRODUÇÃO

1.1 CONTEXTO

A área de gastronomia está em constante crescimento, de acordo com Ardisson Naim Akel, presidente da Jucepar, o crescimento de empresas no setor chega a ser surpreendente. Somente no primeiro semestre de 2017 foram construídas 1098 empresas. Estes números são extraordinários, já que em 2016 inteiro foram criadas 1325 empresas, então a expectativa é que talvez duas mil empresas sejam criadas até o fim do ano.

Para Akel, o principal motivo para esse crescimento seria a maior procura das pessoas por refeições fora de casa. Antigamente comer fora de casa era ir em restaurantes com grande equipe, vários cozinheiros. Hoje há várias outras opções, como *food trucks*, *contêineres*, bares e *pubs*.

Segundo a pesquisa “Preço Médio 2017”, encomendada pela Associação das Empresas de Refeição e Alimentação Convênio para o Trabalho(Assert), o brasileiro está cada vez mais comendo fora de casa, seja por privacidade, conforto ou até um pouco de luxo. O estudo aponta que os brasileiros gastam, em média, R\$ 32,94 com alimentação em restaurantes nas grandes capitais por dia.

Há vários motivos para o brasileiro estar adquirindo o hábito de comer fora. Um dos principais é a urbanização crescente, já que se tem um maior número de horas trabalhadas e de opções de lazer. Além disso, a cultura de ter um membro da família para cuidar da alimentação da casa está diminuindo com a mulher mais ativa no mercado de trabalho e em vez de variar esta tarefa, tanto o homem quanto a mulher preferem a comodidade de comer fora.

Visto que cada vez mais as pessoas comem fora de casa é notório que os estabelecimentos tenham mais clientes e, quanto mais clientes, mais difícil se torna o atendimento aos mesmos e mais fácil a ocorrência de erros como perda de comandas, atraso nos pratos ou até duas reservas para uma mesma mesa.

1.2 JUSTIFICATIVA

Devido a crescente demanda por estabelecimentos do ramo gastronômico, pressupõe que há uma necessidade de uma otimização no atendimento e também na gerência de um restaurante.

A administração manual por diversas vezes pode ser falha, com isso um *software* com mecanismos que auxiliam na gerência de reservas, e que também possibilite um controle mais eficaz e padronizado de pedidos, dentre outras funções que ajudam na logística de um restaurante, com isso impactando positivamente a condução de um estabelecimento gastronômico.

Tendo como motivação a ideia de que cada vez mais pessoas buscam realizar suas refeições em restaurantes, um *software* que possa ajudar um restaurante a ser administrado acaba possibilitando que o gerente do estabelecimento consiga se dedicar mais a atividades de atendimento ao público, com isso aumentando a satisfação de seus clientes.

1.3 PROPOSTA

Propomos o desenvolvimento de um *website* para automatizar o atendimento ao cliente, facilitando por meio de um cardápio digital e um registro de reservas. O funcionário poderá registrar o pedido diretamente no sistema gerando uma comanda virtual evitando, assim, a perda de dados, finalizando o atendimento ele poderá fechar a conta e o sistema calculará o total, disponibilizando a opção de dividir o preço final em um número n de pessoas. O gerente terá todas as funções do funcionário e, também, poderá gerenciar os usuários e fazer alterações no cardápio.

1.4 ORGANIZAÇÃO DO DOCUMENTO

Este documento está organizado em 8 partes, sendo elas: Introdução, Descrição Geral do Sistema, Desenvolvimento do Projeto, Requisitos do Sistema, Análise do Sistema, Implementação, Considerações Finais e Bibliografia. Para uma explicação mais explícita cada uma das partes citadas acima contém algumas subdivisões.

2 DESCRIÇÃO GERAL DO SISTEMA

2.1 OBJETIVOS

2.1.1 Objetivo Geral

O objetivo deste projeto é otimizar e facilitar o gerenciamento de restaurantes através de uma ferramenta *web* que será desenvolvida para automatizar serviços que antes eram feitos manualmente.

2.1.2 OBJETIVOS ESPECÍFICOS

- Gerenciar reservas
- Apresentar e gerenciar pedidos
- Auxiliar o fechamento de contas

2.2 LIMITES E RESTRIÇÕES

- Não terá suporte 24 horas pelos desenvolvedores;
- Sem atualizações semanais, apenas mensais.
- O sistema não irá fazer possibilitar o cadastro de clientes dos restaurantes.
- Os únicos usuários que poderão fazer alterações no sistema serão o(s) funcionário(s) e o(s) gerente(s).

2.3 DESCRIÇÃO DOS USUÁRIOS DO SISTEMA

Os atores que serão envolvidos na solução serão cliente, que participa diretamente na reserva de mesas, no pedido e também no pagamento, o atendente que participa da reserva de mesas e do pagamento, o garçom que cuida apenas dos pedidos feitos pelos clientes e o gerente que, além das funcionalidades dos outros empregados ele também irá gerenciar os funcionários e também irá poder administrar o cardápio. O sistema será destinado à empresas do ramo gastronômicos, tais como restaurantes, lanchonetes, bistrôs, etc., tendo como clientes empresas que tenham o interesse em investir no ramo gastronômico.

3 DESENVOLVIMENTO DO PROJETO

3.1 TECNOLOGIAS E FERRAMENTAS

- API: JDBC - Fazer envio de instruções SQL para o banco de dados relacional
- Banco de dados: Postgresql - Sistema gerenciador de banco de dados relacional
- Confecção dos documentos (diagramas): Astah - Utilizado para criação e edição de diagramas da UML
- Criação do cronograma: GanttProject - Organizador de tarefas e cronogramas do projeto
- *Framework* front-end: BootStrap - Tecnologia para ajudar na estilização do layout
- *Framework* back-end: JSP - Tecnologia para ajudar no desenvolvimento web dinâmico.
- Hospedagem: 000webhost - Servidor de hospedagem
- Linguagem: HTML 5, css3, javascript,java EE - Linguagem de programação adotada
- Modelo Relacional: Workbench - Ferramenta para modelagem do banco de dados relacional
- Prototipação de telas: Photoshop - Ferramenta para edição de imagens
- Repositório: GitHub - Utilizado para repositório de códigos e controle de versão

3.2 METODOLOGIA DE DESENVOLVIMENTO

A metodologia para o desenvolvimento deste sistema que foi escolhida é o *Scrum*, pois com o aumento da interação do *Product Owner* com os desenvolvedores do software conforme as *Sprints* são apresentadas e o *Product Backlog* vai se desenvolvendo, é possível um aprimoramento no *software* durante a fabricação do mesmo. No *Daily Scrum* é apresentado o que ocorreu na *Sprint*

anterior com isso a interação entre os desenvolvedores será ampliada de uma forma que será possível aumentar a eficiência e eficácia do *software* em conjunto ao *Product Owner*.

3.3 CRONOGRAMA PREVISTO

ID	ATIVIDADE	DURAÇÃO	INÍCIO	FIM
1	Diagramas de caso de uso	1 semana	04/09/2017	12/09/2017
2	Banco de dados	1 semana	18/09/2017	22/09/2017
3	Diagrama de classe	3 dias	18/09/2017	20/09/2017
4	Diagrama de atividades	2 dias	21/09/2017	22/09/2017
5	Protótipos de telas	1 semana	18/09/2017	22/09/2017
6	Desenvolvimento	4 semanas	25/09/2017	20/10/2017
7	Testes	3 semanas	09/09/2017	27/10/2017
8	Finalização	3 dias	30/10/2017	01/11/2017

ID	ATIVIDADE	RESPONSÁVEL
1	Diagramas de caso de uso	João Victor
2	Banco de dados	João Victor
3	Diagrama de classe	Filipe
4	Diagrama de atividades	Filipe
5	Protótipos de telas	Rafael
6	Desenvolvimento	João Victor, Filipe e Rafael
7	Testes	João Victor, Filipe e Rafael
8	Finalização	João Victor, Filipe e Rafael

4 REQUISITOS DO SISTEMA

4.1 REQUISITOS FUNCIONAIS

ID	FUNCIONALIDADE	PRIORIDADE
1	Gerenciar reservas de mesas	Essencial
2	Consultar reserva de mesas	Essencial
3	Disponibilizar cardápio	Essencial
4	Gerenciar pedidos	Essencial
5	Calcular conta	Importante
6	Gerenciar cardápio	Essencial
7	Gerenciar usuários	Essencial

1. Possibilitará o gerenciamento de reservas de mesas, será possível alterar, cadastrar e remover reservas.
2. Disponibilizará a consulta das mesas reservadas.
3. Mostrará os itens disponíveis para o consumo do restaurante.
4. Permitirá o gerenciamento de pedidos, será possível alterar, cadastrar e remover pedidos.
5. Oferecerá um mecanismo que irá calcular o valor da conta.
6. Haverá uma forma de cadastramento de itens no cardápio, remoção e também alteração.
7. Proporcionará o gerenciamento de funcionários, cadastrar, alterar e também remover funcionários.

4.2 REQUISITOS NÃO-FUNCIONAIS

ID	REQUISITO	CATEGORIA
1	Interface amigável	Usabilidade
2	Tempo de resposta rápido	Padronização
3	Ser manuseado por vários usuários concorrentemente	Usabilidade

1. Promover uma interface limpa e de fácil manuseio.
2. Possibilitar que o sistema não demore mais de três segundos para realizar a ação requisitada.
3. Proporcionar que diversos usuários consigam acessar a ferramenta concorrentemente.

4.3 DIAGRAMAS DE CASOS DE USO

4.3.1 NÍVEL 0

O diagrama a seguir apresenta as interações que o sistema era proporcionar para cada usuário (Cliente, Funcionário, Gerente).

Os Casos de Uso apresentados na imagem, têm as seguintes especificações:

- Consultar Pedido - Caso solicitado esse método irá apresentar o andamento do pedido.
- Consultar Reserva - Caso solicitado esse método irá apresentar os dados das reservas, tão como os horários disponíveis para agendamento.
- Gerenciar Pedidos - Esse método será capaz de excluir, adicionar e alterar pedidos, desde que o mesmo não esteja sendo preparado.
- Fechar Conta - Esse método mostrará o valor total da conta, também será atribuído a opção de dividir o valor total por uma quantidade “n” de pessoas.
- Gerenciar Reservas - Esse método será capaz de excluir, adicionar e alterar reservas.
- Gerenciar Cardápio- Esse método será capaz de excluir, adicionar e fazer alterações nos itens apresentados no cardápio.
- Gerenciar Usuários - Esse método será capaz de excluir e adicionar novos usuários, também será possível alterar o “tipo” de usuário.

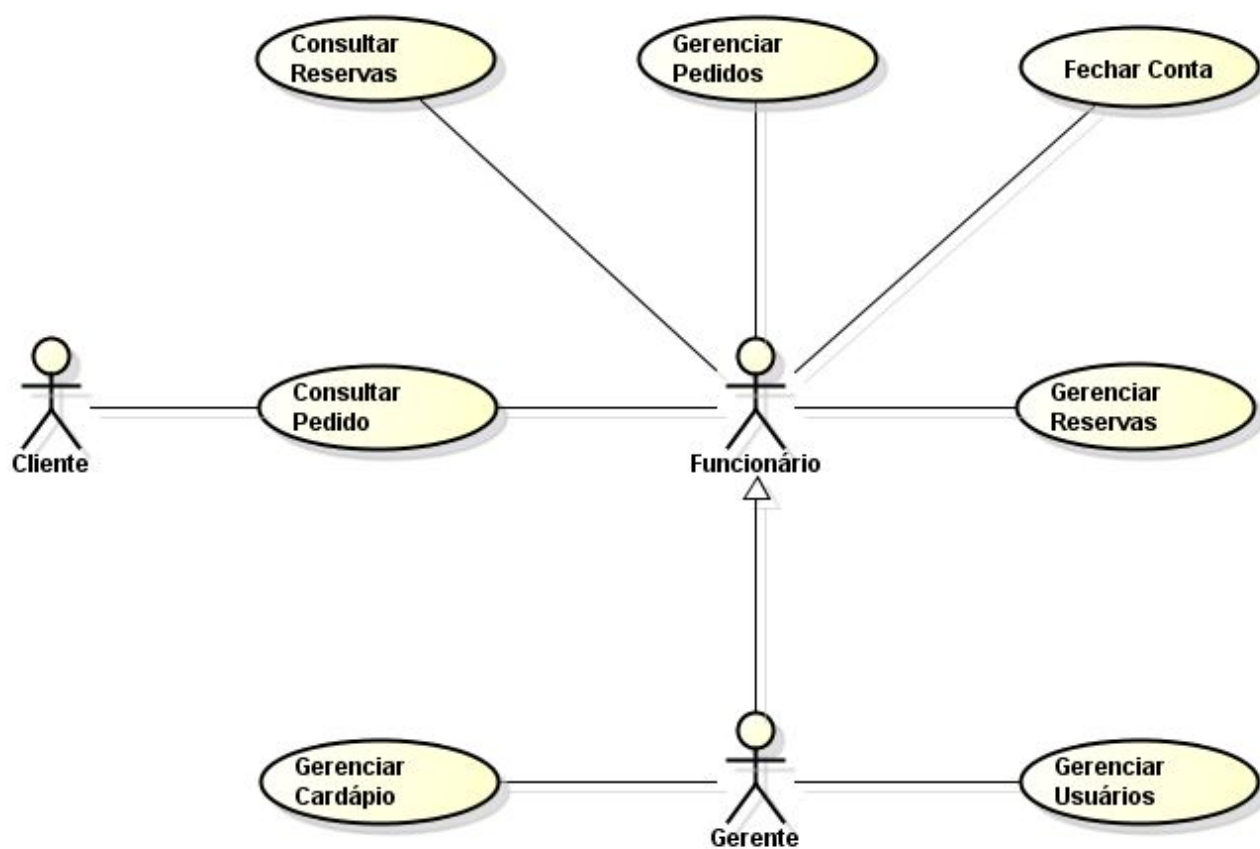


Diagrama de Casos de Uso Nível 0 - Visão geral das funções do software.

4.3.2 GERENTE

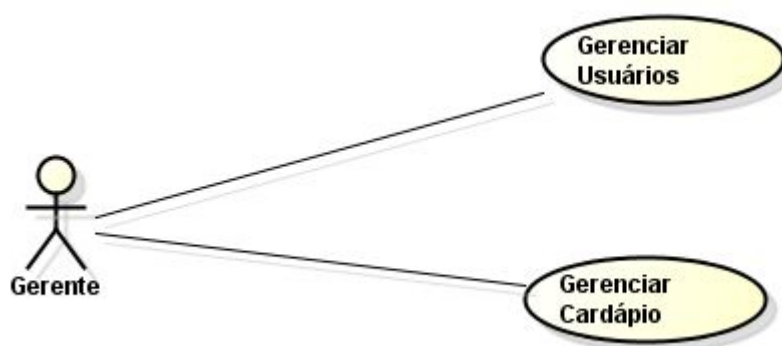


Diagrama de Casos de Uso das funções não herdadas do usuário do "tipo" Gerente.

4.3.3 FUNCIONÁRIO

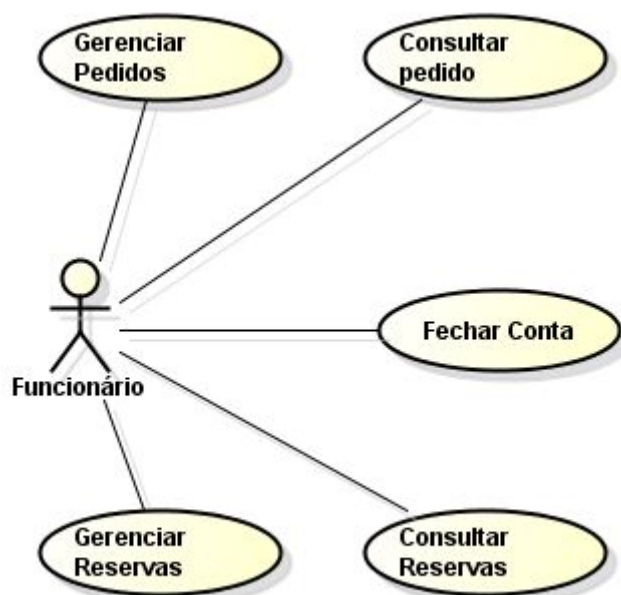
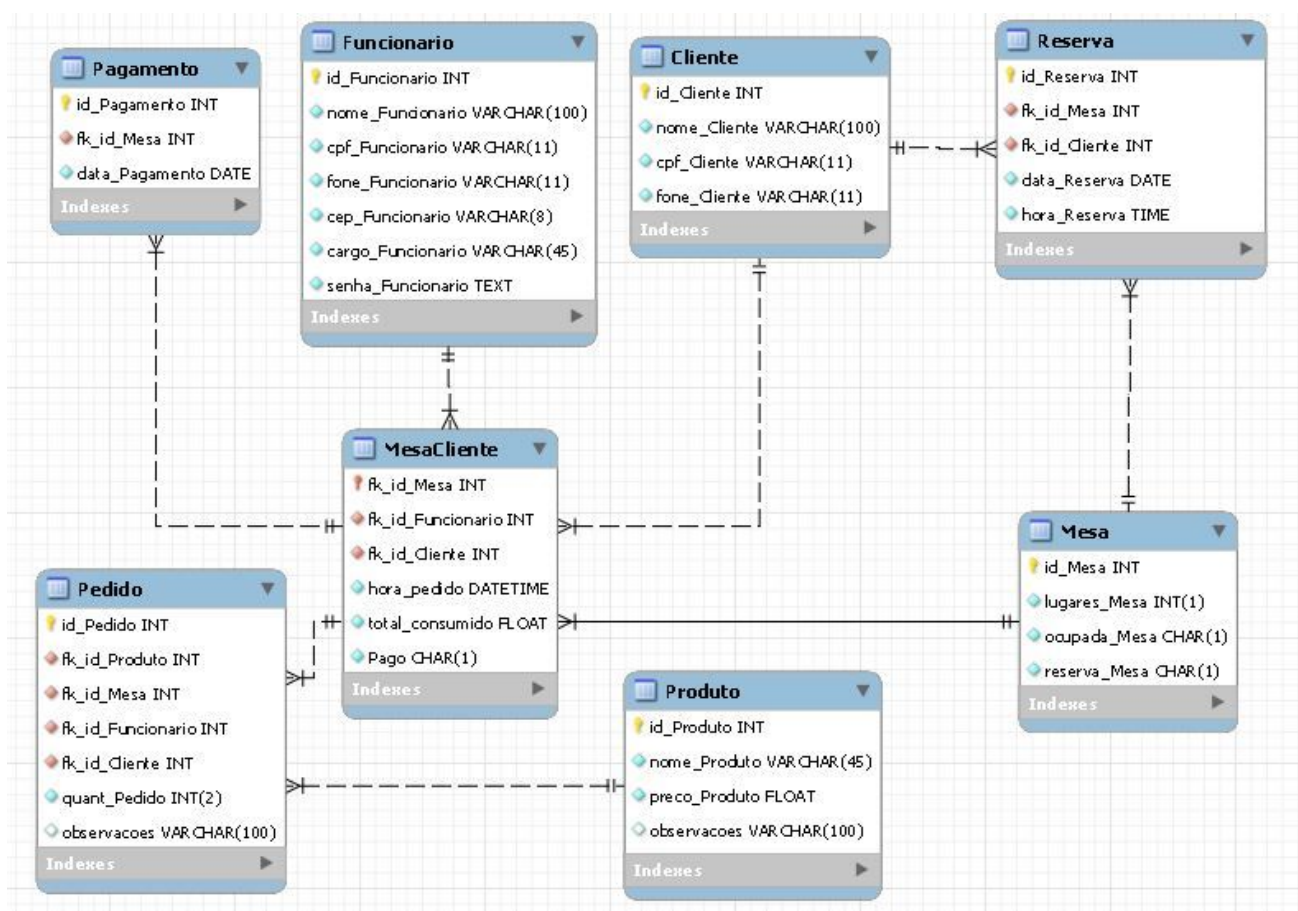


Diagrama de Casos de Uso das das funções do usuário do "tipo" Funcionário.

5 ANÁLISE DO SISTEMA

5.1 MODELO DO BANCO DE DADOS

5.1.1 MODELO CONCEITUAL



Modelo conceitual e lógico.

5.1.2 MODELO LÓGICO

Modelo lógico apresentado no item anterior..

5.1.3 DICIONÁRIO DE DADOS

Tabela: Funcionário Descrição: Armazena os dados dos funcionários			
CAMPO	TIPO	OBSERVAÇÃO	RESTRIÇÃO
id_Funcionario	int	Constitui a chave primária	Não contém restrição
nome_Funcionario	varchar(100)	Não nulo	Deve conter 100 caracteres
cpf_Funcionario	varchar(11)	Não nulo	Deve conter exatamente 11 caracteres
fone_Funcionario	varchar(11)	Não nulo	Deve conter 11 caracteres
cep_Funcionario	varchar(8)	Não nulo	Deve conter 45 caracteres
cargo_Funcionario	varchar(45)	Não nulo	Deve conter 45 caracteres
senha_Funcionario	text	Não nulo	Não contém restrição

Tabela: Pagamento Descrição: Armazena os dados dos pagamento			
CAMPO	TIPO	OBSERVAÇÃO	RESTRIÇÃO
id_Pagamento	int	Constitui a chave primária	Não contém restrição
id_Mesa	int	Constitui a chave estrangeira	Não contém restrição
data_Pagamento	date	Não nulo	Não contém restrição

Tabela: Cliente Descrição: Armazena os dados dos clientes			
CAMPO	TIPO	OBSERVAÇÃO	RESTRIÇÃO
id_Cliente	int	Constitui a chave primária	Não contém restrição
nome_Cliente	varchar(100)	Não nulo	Deve conter 100 caracteres
cpf_Cliente	varchar(11)	Não nulo	Deve conter exatamente 11 caracteres
fone_Cliente	varchar(11)	Não nulo	Deve conter 11 caracteres

Tabela: Mesa Descrição: Armazena os dados das mesas			
CAMPO	TIPO	OBSERVAÇÃO	RESTRIÇÃO
id_Mesa	int	Constitui a chave primária	Não contém restrição
lugares_Mesa	int(1)	Não nulo	Deve conter 1 dígito

ocupada_Mesa	char(1)	Não nulo	Deve conter 1 caractere
reservado_Mesa	char(1)	Não nulo	Deve conter 1 caractere

Tabela: Produto**Descrição: Armazena os dados dos produtos**

CAMPO	TIPO	OBSERVAÇÃO	RESTRIÇÃO
id_Produto	int	Constitui a chave primária	Não contém restrição
nome_Produto	varchar(45)	Não nulo	Deve conter 45 caracteres
preco	float	Não nulo	Não contém restrição
observacoes	varchar(100)	Sem observação	Deve conter 100 caracteres

Tabela: Pedido**Descrição: Armazena os dados dos pedidos**

CAMPO	TIPO	OBSERVAÇÃO	RESTRIÇÃO
id_Pedido	int	Constitui a chave primária	Não contém restrição
fk_id_Produto	int	Constitui uma chave estrangeira	Não contém restrição
fk_id_Mesa	int	Constitui uma chave estrangeira	Não contém restrição
fk_id_Funcionario	int	Constitui uma chave estrangeira	Não contém restrição
fk_id_Cliente	int	Constitui uma chave estrangeira	Não contém restrição
quant_Pedido	int(2)	Não nulo	Deve conter 2 dígitos
observacoes	varchar(100)	Sem observação	Deve conter 100 caracteres

Tabela: Reserva**Descrição: Armazena os dados das reservas**

CAMPO	TIPO	OBSERVAÇÃO	RESTRIÇÃO
id_Reserva	int	Constitui a chave primária	Não contém restrição
fk_id_Mesa	int	Constitui uma chave estrangeira	Não contém restrição
fk_id_Cliente	int	Constitui uma chave estrangeira	Não contém restrição
data_Reserva	date	Não nulo	Não contém restrição

hora_Reserva	time	Não nulo	Não contém restrição
--------------	------	----------	----------------------

Tabela: MesaCliente			
Descrição: Armazena os dados da mesa reservada			
CAMPO	TIPO	OBSERVAÇÃO	RESTRIÇÃO
fk_id_Mesa	int	Constitui uma chave primária estrangeira	Não contém restrição
fk_id_Funcionario	int	Constitui uma chave estrangeira	Não contém restrição
fk_id_Cliente	int	Constitui uma chave estrangeira	Não contém restrição
hora_pedido	datetime	Não nulo	Não contém restrição
total_Consumido	float	Não nulo	Não contém restrição
pago	char(1)	Não nulo	Deve conter 1 caractere

5.2 DIAGRAMA DE CLASSES

O diagrama a seguir representa a estrutura e as relações das classes que serão implementadas no sistema.

O cliente solicitará reserva e realizará pedidos, a relação “muitos-para-muitos” entre Pedido e Produto irá criar uma classe classificada como itens_pedido. A realização de pedidos gera a conta, que só será finalizada com o pagamento da mesma.

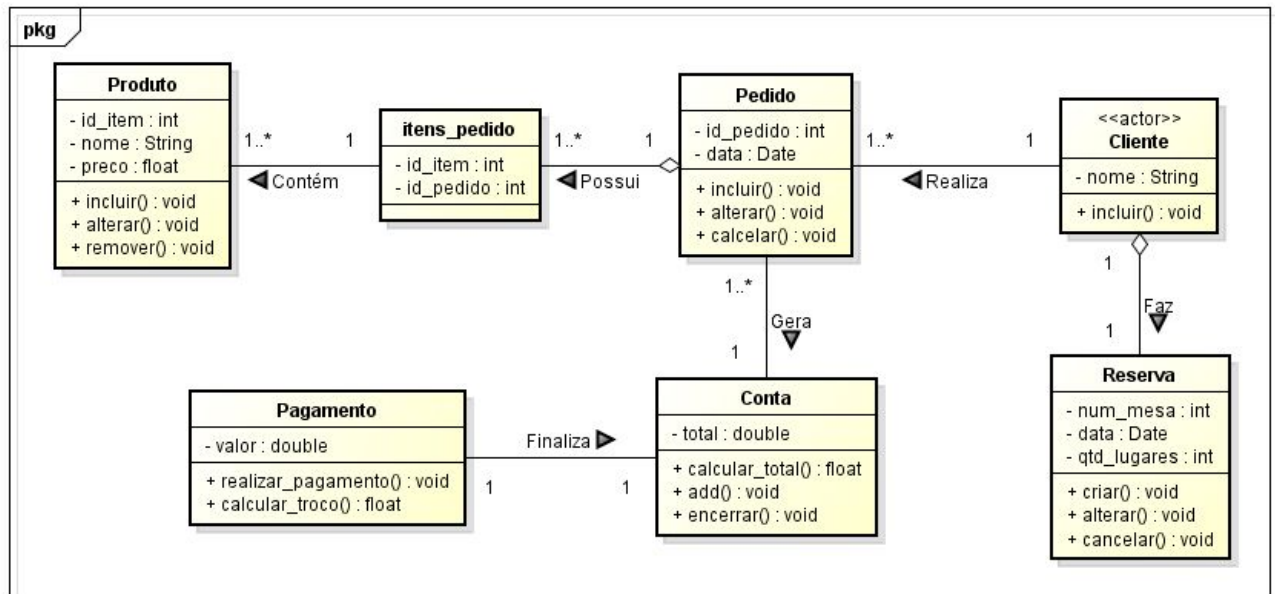


Diagrama de classe das funções gerais do sistema.

5.3 DIAGRAMA DE ATIVIDADES

5.3.1 GERENCIAR RESERVA

No diagrama de atividades a seguir será apresentado as funções incluídas em Gerenciar Reserva, esse método poderá consultar, alterar, excluir e também criar novas reservas com a seguinte sequência estruturada (passo-a-passo):

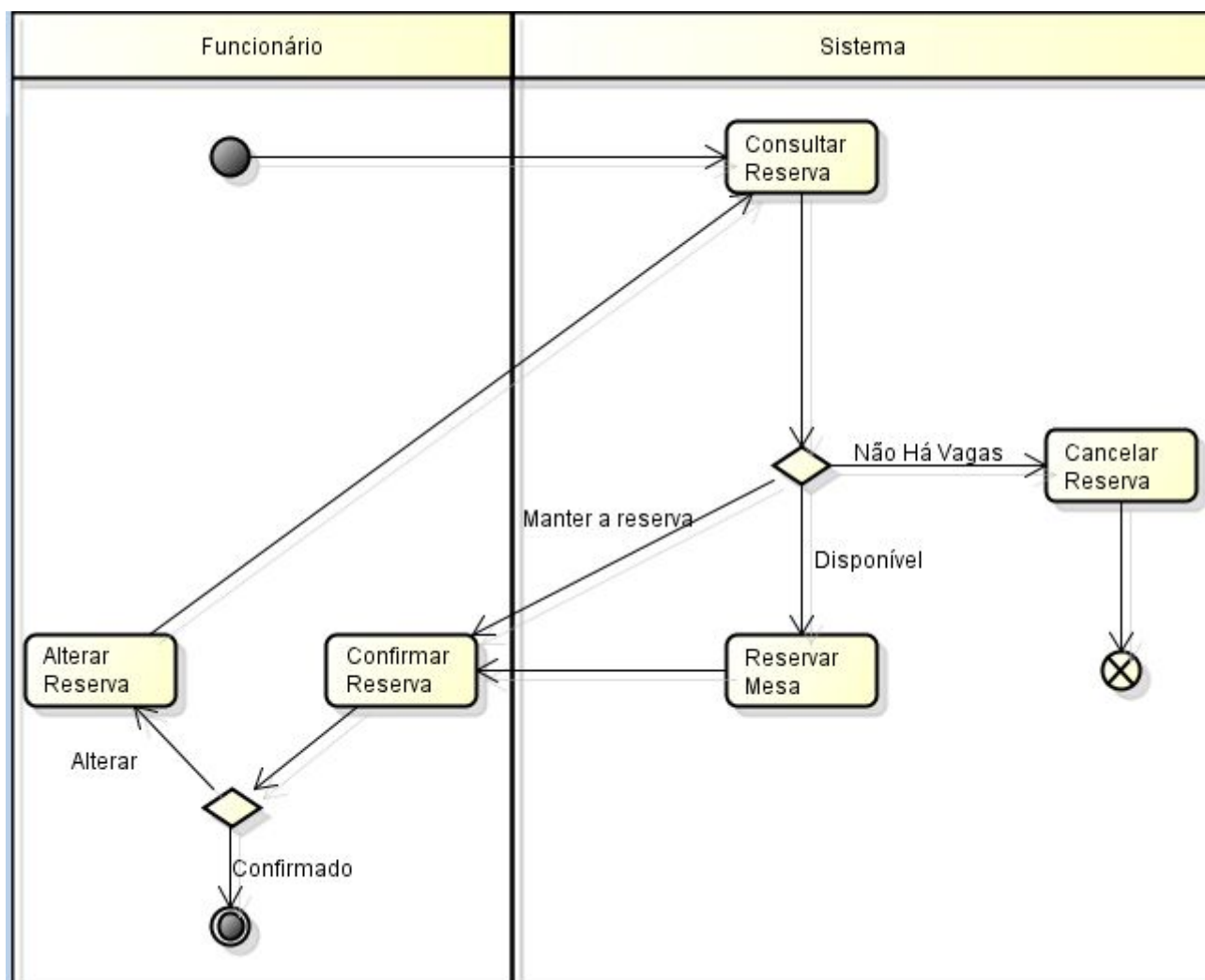


Diagrama de Atividades do método Gerenciar Reserva.

5.3.2 GERENCIAR PEDIDO

Esse diagrama de sequência apresenta como funcionará as funções de consultar, incluir, alterar e cancelar um pedido.

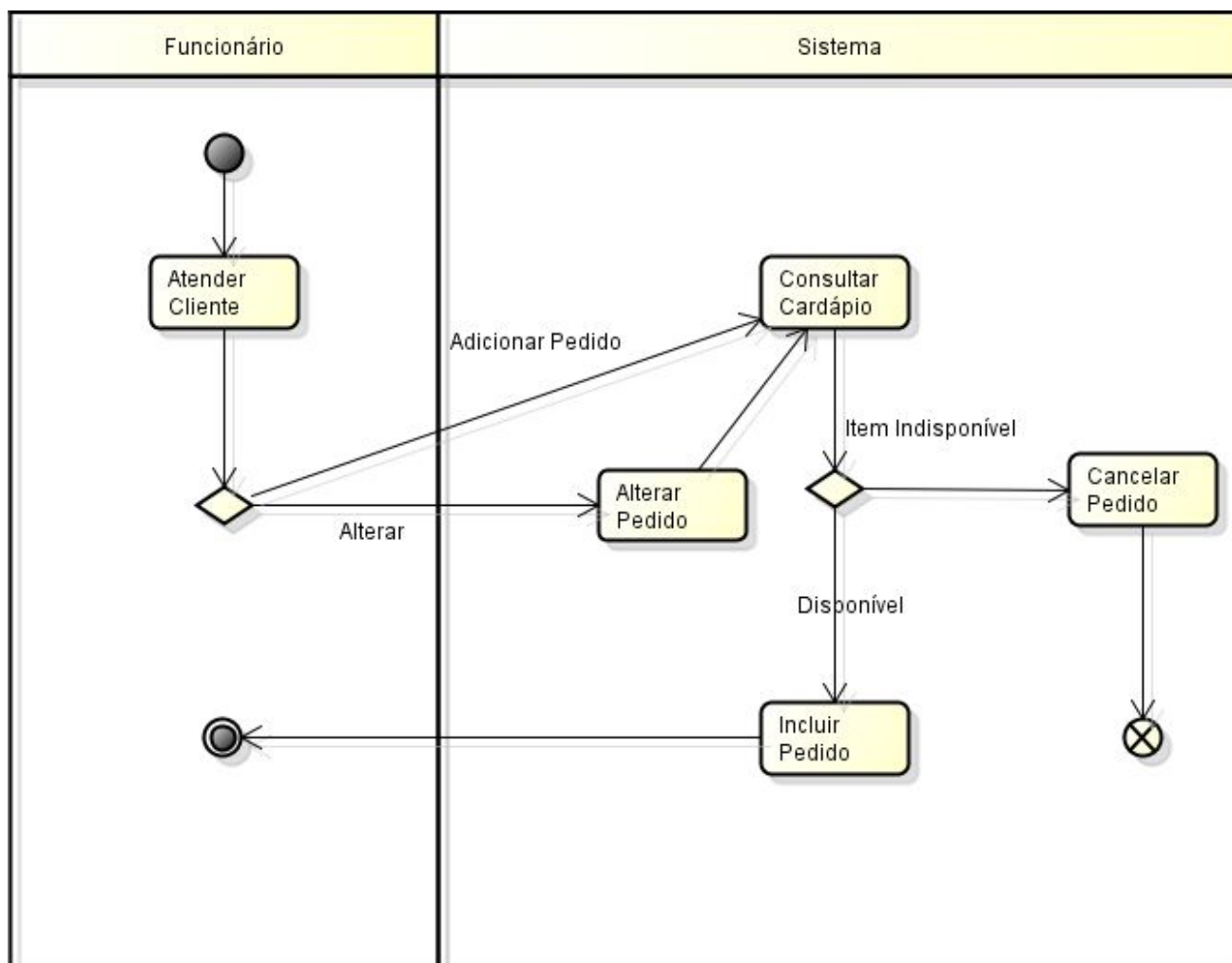


Diagrama de Atividades do método Gerenciar Pedido.

5.3.3 FECHAR CONTA

Esse diagrama de sequência apresenta como funcionará a função de encerrar uma conta no restaurante.

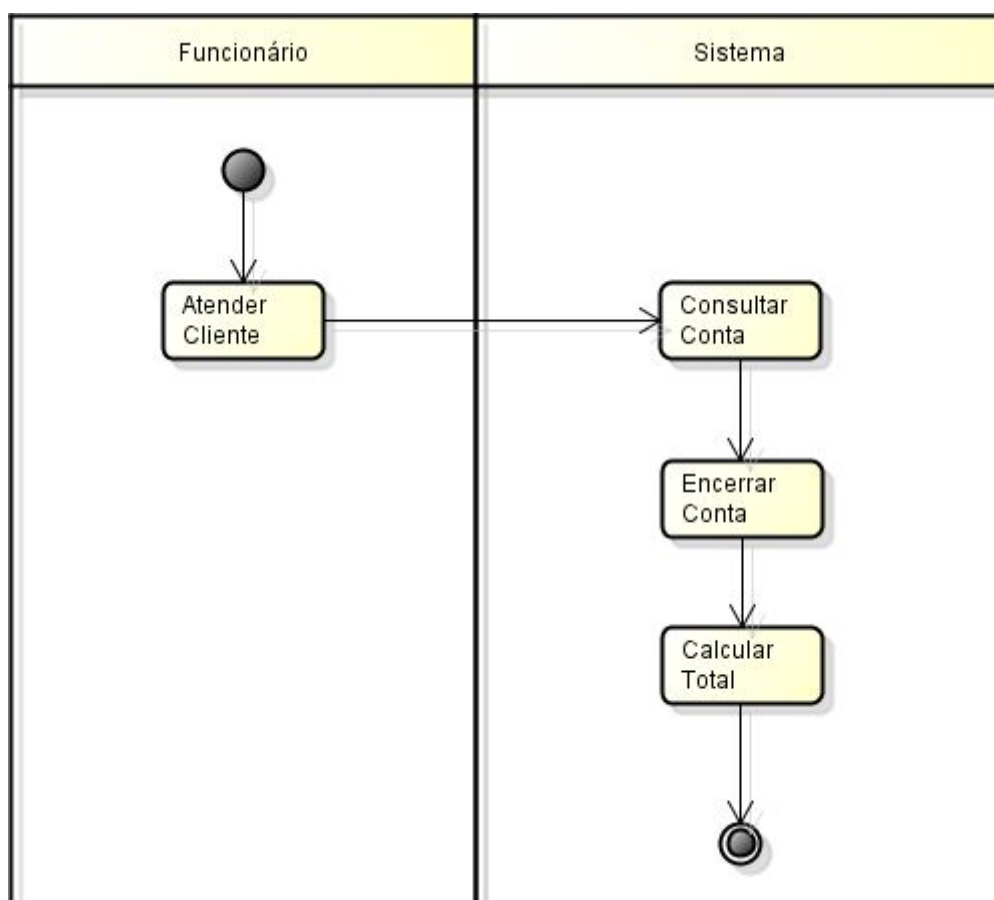


Diagrama de Atividades do método Calcular Conta.

5.3.4 GERENCIAR USUÁRIO

Neste diagrama de atividades é exposto as funções do método Gerenciar Usuários, nele é apresentado as funções consultar, alterar, excluir e cadastrar usuário. Caso chegue na fase que é necessária a confirmação do cadastro (seja por criação ou alteração) e seja desejado fazer uma alteração ao invés de concluir o cadastro/alteração o diagrama de atividades mostra que o método volta para a fase de consulta de usuários.

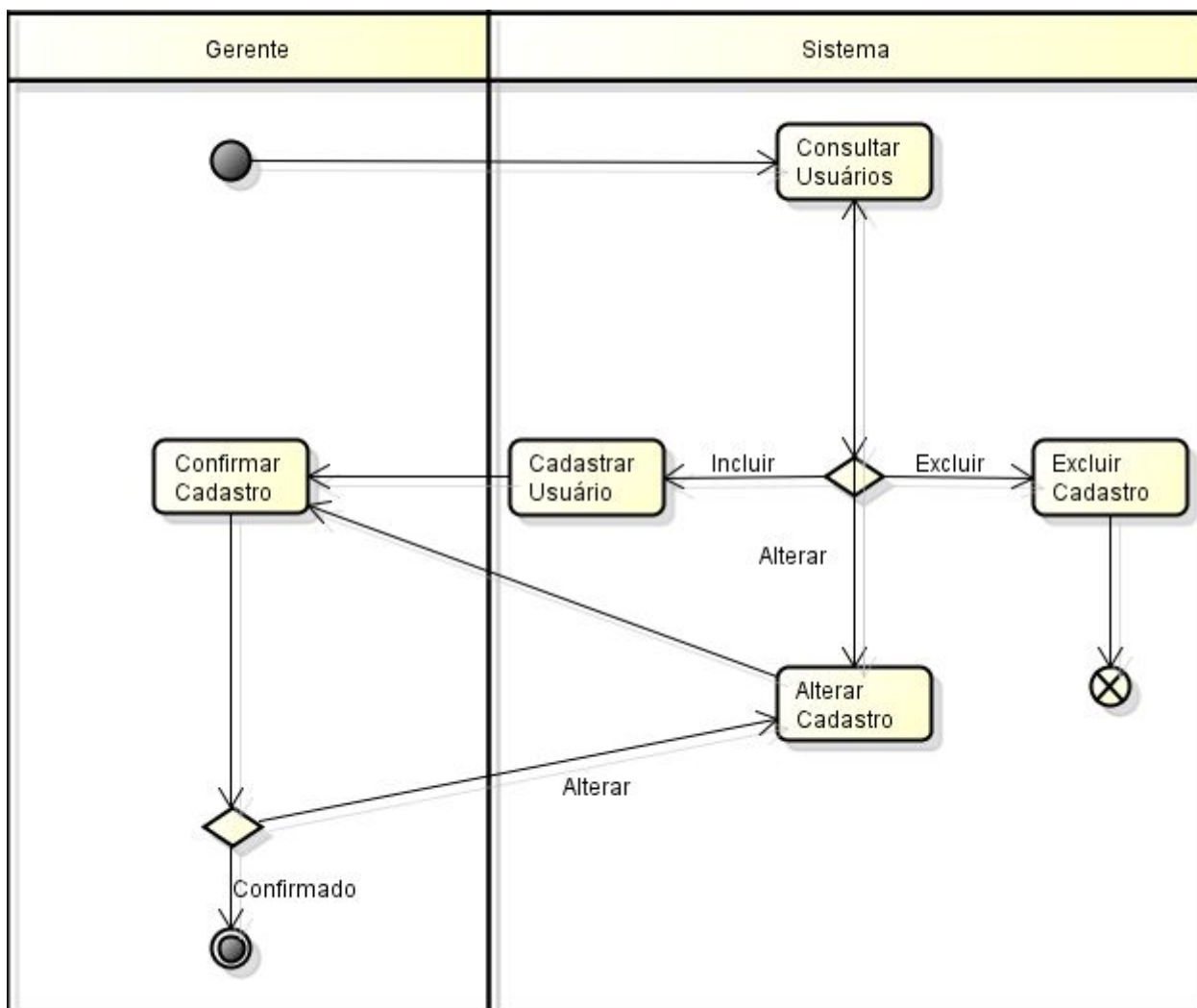


Diagrama de Atividades do método Gerenciar Usuário.

5.3.5 GERENCIAR CARDÁPIO

O diagrama a seguir segue o padrão do diagrama “Gerenciar Usuário”.

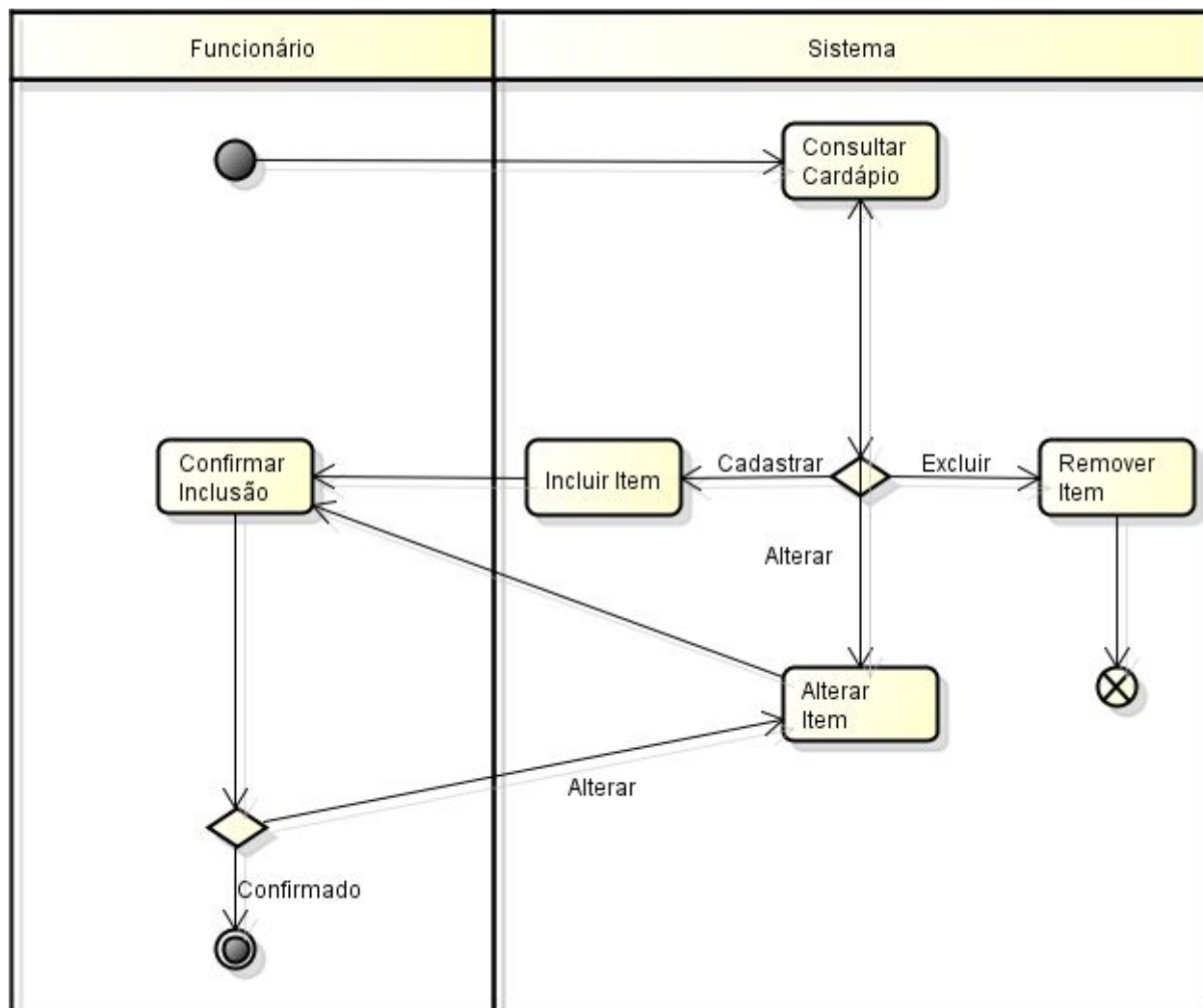
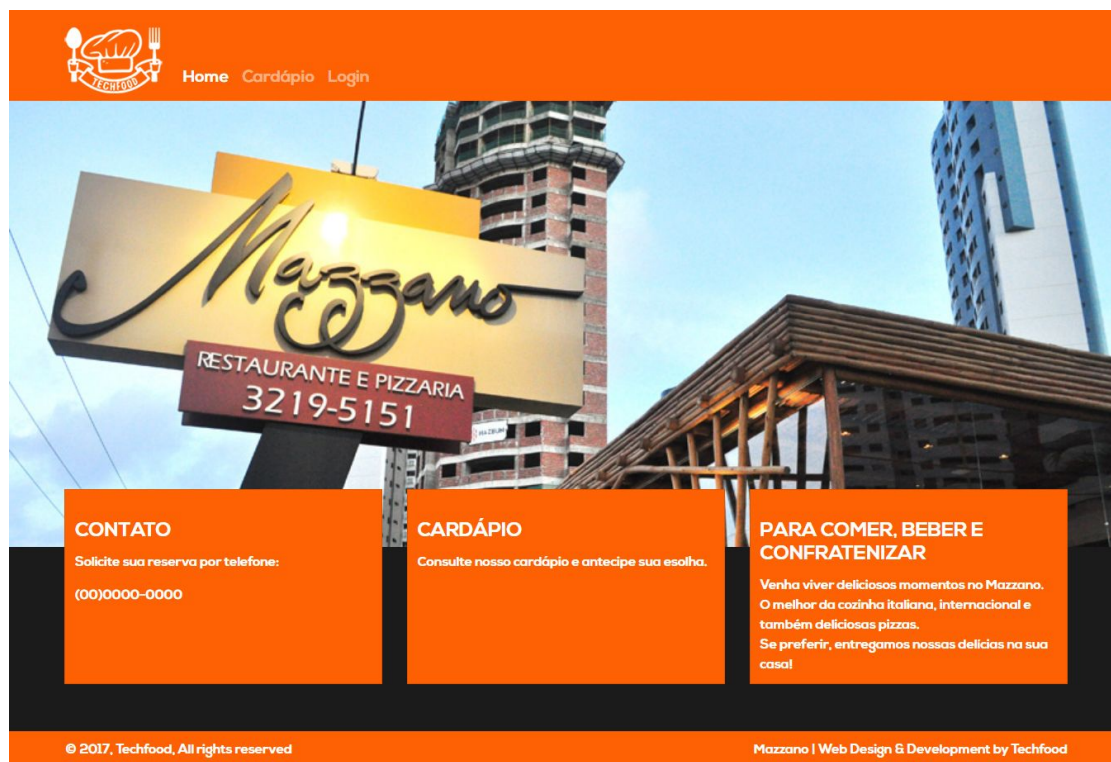


Diagrama de Atividades do método Gerenciar Cardápio.

6 IMPLEMENTAÇÃO

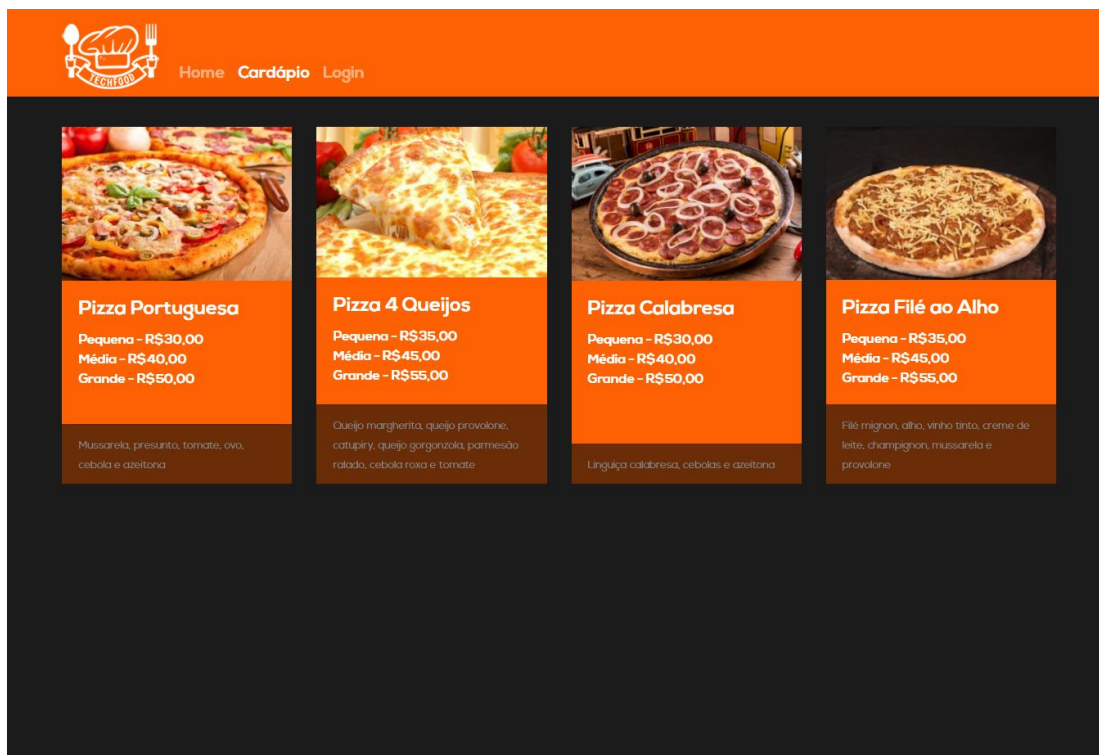
6.1 PROTÓTIPOS DE TELAS

6.1.1 TELA INICIAL



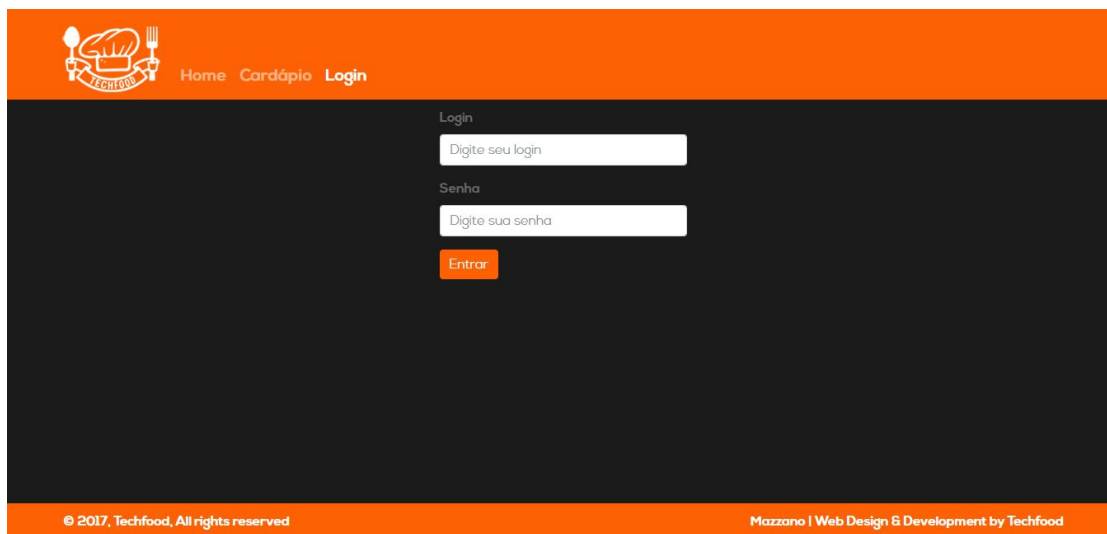
Protótipo da tela inicial tendo como base uma pizzaria.

6.1.2 CARDÁPIO



Protótipo da tela do cardápio tendo como base uma pizzaria.

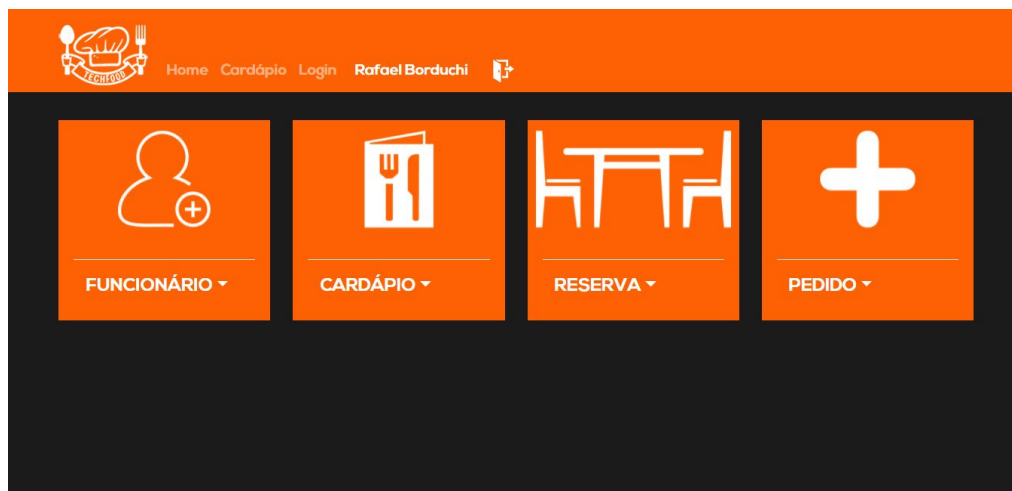
6.1.3 LOGIN



Protótipo da tela de login.

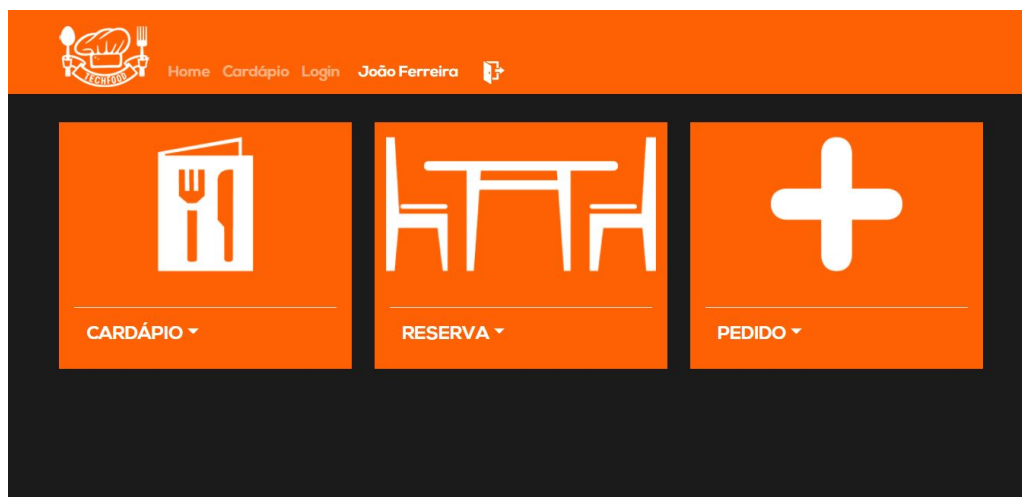
6.1.4 USUÁRIO

6.1.4.1 GERENTE



Protótipo da tela do funcionário acessada pelo gerente.

6.1.4.2 FUNCIONÁRIO

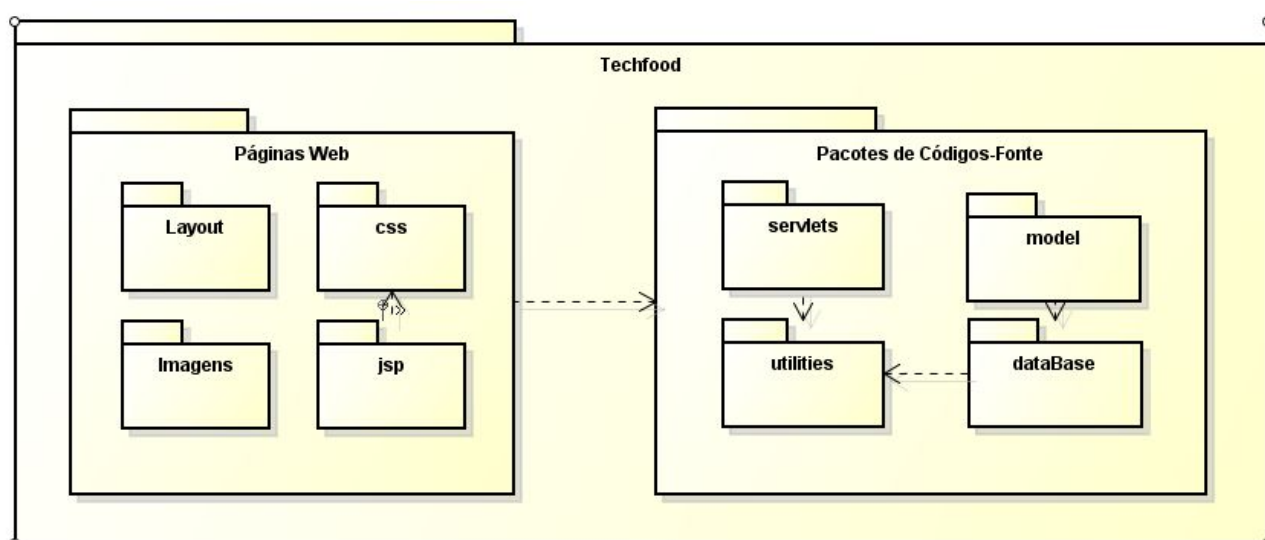


Protótipo da tela do funcionário acessada pelo funcionário.

6.2 DESCRIÇÃO DO CÓDIGO

A estrutura do projeto foi se desenvolvendo conforme a evolução do mesmo, devido a isso, sua estrutura ficou personalizada de forma que a equipe pudesse trabalhar melhor. Nesta personalização é inicialmente dividido o código em pastas, ao adentrar nelas é possível encontrar presença de pacotes e subpastas para assim conseguir dividir o código em mais áreas.

A seguir segue o diagrama de pacotes do nosso projeto apresentando pastas e pacotes, foram descartadas às pastas “Bibliotecas” e “Arquivos de Configuração”, e também às subpastas padrões que se encontram na pasta “Pacotes de Códigos-Fontes”, essas pastas são a “META-INF” e “WEB-INF”.



Nas pastas que estão dentro da pasta “Páginas Web” é utilizado o padrão de nomenclatura “Camel Case”, neste padrão os arquivos têm a primeira letra minúscula e nos arquivos, objetos e métodos que contém mais de uma palavra, as palavras seguintes começam com a letra maiúscula, não são utilizados acentos ou espaços, mas é aceito abreviações.

Dentro de “Páginas Web” são encontradas as pastas “Imagens”, “Layout”, “jsp” e a pasta “css”. Dentro da “Imagens” foram postas as imagens utilizadas no projeto, do mesmo modo a pasta “Layout” abriga o layout desenvolvido para nosso software. Dentro da pasta “css” é encontrado o css de nosso website, também é

encontrado a pasta “jsp” onde ficam as páginas secundárias que desenvolvemos para o projeto e por fim também dentro de “Páginas Web” encontrasse a página inicial do nosso website.

Dentro da pasta “Pacotes de Códigos-Fonte” são encontradas os pacotes “servlets”, “model”, “utilities” e “dataBase”, o padrão de nomenclatura utilizado é o “Pascal Case”, nele as classes começam com a primeira letra maiúscula e a primeira letra de cada palavra concatenada também. Os métodos e objetos contidos dentro desta pasta utilizam o padrão de nomenclatura “Camel Case” que fora explicado anteriormente. Não são utilizados acentos e espaços, mas abreviações são aceitas.

No pacote “servlets” é encontrado às classes e métodos que controlam às classes encontradas na pasta model. No pacote “dataBase” é encontrado a classe responsável pela conexão com o banco de dados de nosso sistema. Por fim no pacote “utilities” apresenta uma classe geradora de códigos.

A seguir é apresentado o Diagrama de Componentes, este diagrama tem como funcionalidade apresentar o conjunto de interfaces de nosso *software*, nele é mostrado as interfaces e classes do sistema.

A imagem expõe às requisições necessárias e por quem elas devem ser feitas para o funcionamento do *software*. A seta com linha pontilhada representa a classe ou *interface* que solicita a requisição, enquanto a seta seguida de uma elipse preenchida representa a classe ou *interface* que fornece os dados requisitados.

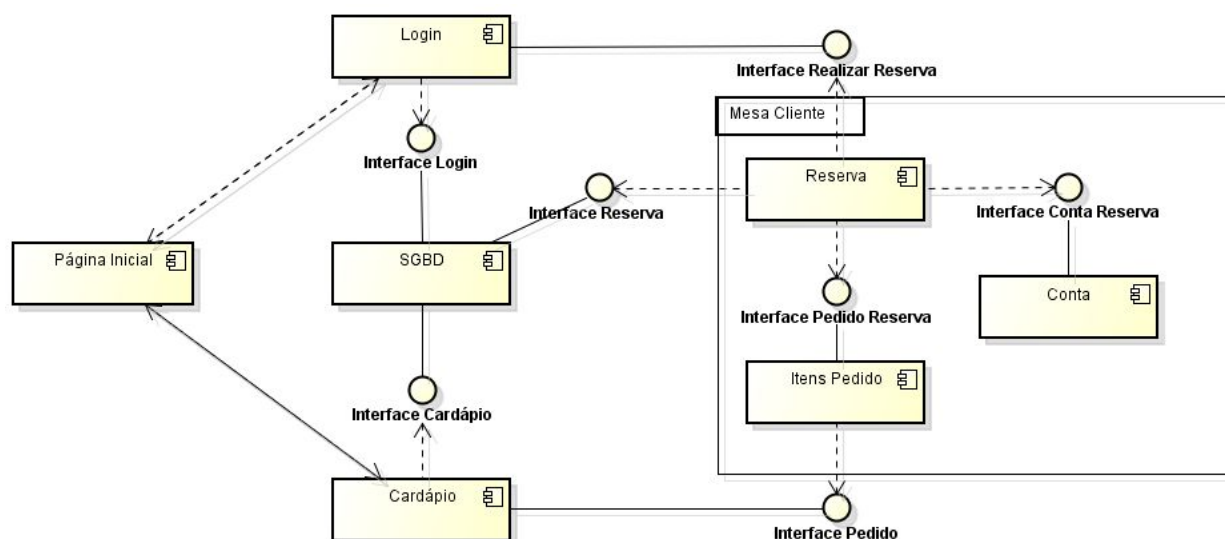


Diagrama de Componentes, apresentando as ligações entre os componentes do software.

7 CONSIDERAÇÕES FINAIS

Estando perto da entrega da versão final de nosso projeto, conseguimos obter diversos conhecimentos no ramo de desenvolvimento de *software*, sendo os principais deles o aprimoramento em linguagens de programação, a importância do planejamento da concepção de um *software* e o mais importante fora a melhora na habilidade de buscar conhecimento.

Tendo em vista que o “Techfood” foi o primeiro software que desenvolvemos desde sua concepção à sua implementação, encontramos diversas dificuldades. As principais delas foram o conhecimento sobre as tecnologias a utilizar, desde qual a melhor até o próprio domínio da mesma, também vimos a dificuldade na elaboração do *design* e da documentação de um *software*. Acreditamos que a principal dificuldade encontrada foi na administração do tempo, visto que foi a primeira experiência da equipe gerando um *software* dentro de padrões de projetos de *software* e vimos a relevância do cronograma quando bem elaborado.

Nossa equipe a princípio acreditou que a experiência adquirida em sala de aula seria o suficiente para a realização do projeto, supomos que esta tenha sido nossa maior falha. Tendo isso em mente observamos que deve ser acrescentado

no cronograma a contagem de tempo de treinamento para utilização das tecnologias, ato crucial para o sucesso da atividade.

A versão do “Techfood” entregue consegue validar o objetivo específico que propõe apresentar e gerenciar pedidos. A respeito dos requisitos funcionais apenas dois não foram validados, sendo eles o requisito que garantiria a consulta de mesas reservadas e o requisito que gerenciaria essas mesas.

O *software* foi entregue mas não fora concluído, nossa equipe pretende futuramente realizar os casos testes, implementar o objetivo específico de gerenciar reservas e também o de auxiliar o fechamento de contas. Também pretendemos realizar os requisitos funcionais ditos no parágrafo anterior. Por fim pensamos em adicionar padrão *Model-View-Controller (MVC)* em nosso projeto e também adicionar melhorias à interface gráfica.

8 REVISÃO BIBLIOGRÁFICA

ENTENDA O QUE É PASCAL CASE E CAMEL CASE, Disponível em <<http://fabiocabral.gabx.com.br/2013/09/entenda-o-que-e-pascal-case-e-camel-case.html>>. Acesso em: 22 de nov. de 2017.